Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Online Identification of Multivariable Nonlinear Processes with Iterative Structure Learning and Application to a Diesel Engine

Marek Kowalczyk, Boliang Yi, Rolf Isermann *

* *Institute of Automatic Control and Mechatronics, Laboratory of Control Engineering and Process Automation, Technical University Darmstadt, Germany (e-mail: mkowalczyk@iat.tu-darmstadt.de)*

Abstract: This contribution presents a new methodology for the online identification, which allows the modeling of the stationary and dynamic behavior of nonlinear combustion engines with many input and output variables in short time with an appropriate model structure. The used models are local polynomial model trees. The necessary enhancements and adaptations of the identification algorithm for local polynomial model trees for the online methodology are presented. To enable an online identification, not only the parameters but also the structure of the models has to be adapted to the ongoing measuring procedure at a test bench. This structure adaptation methods use iteratively recorded measured data for the determination of an optimized model partition and regressor selection regarding the quality and complexity of the model. This maximizes the iterative improvement of the mathematical model, leading to a reduced test bench time. The applicability of the developed methodology is shown for the identification of a model from both, an artificial test function as well as a real diesel engine.

Keywords: Engine modeling; Mechatronic combustion engine; Iterative measurements; Identification methods; Online identification; Model structure adaptation.

1. INTRODUCTION

1.1 Online measurements

The procedure of conventional engine modeling is performed in individual steps and with an alternation between preparations and analysis at the office and measurement at the test bench. This results usually in repetitions and includes returns e.g. for the design of experiments after model analysis if too few design points in the relevant range have been measured. Such repetitions require additional test bench time.

A significant improvement regarding the resulting measurement time, analysis effort and quality of the models can be achieved with an coupled (online) measurement and modeling procedure (Kowalczyk [2013]). The online procedure operates such that the results of the preceding step are used for the design of the next step. For example model attributes, like e.g. the relevant regressors of the model, can be adaptated based on the newly measured data.

Through the combination of the measurement actuation with real-time hardware and data evaluation with a personal computer, a parallelization of the measurement and the modeling process can be achieved as shown in Fig.1, beginning with the 2^{nd} iteration. With the help of the parallelization, the conversion and analysis of the data of the previous measurement iteration as well as model identification and analysis can be performed during the actual running measurement. This makes an evaluation afterward in the office (as in the classical approach) unnecessary, saving the time needed for it. The parallelization requires, especially for dynamic modeling, that the evaluation is faster than the parallel measurement. Otherwise, standstill periods occur, leading to poorly excited signals for the identification process. A further advantage of the resulting iterative batch processing is the ability to use the ideas of offline analysis and identification methods (e.g. regressor selection, net structure adaptation).

The online procedure is characterized by the use of an automatic iterative approach, reduced user interventions during the measurement and modeling process as well as direct data processing, model identification and validation at the test bench. Furthermore an automatic actuation, data sampling, data analysis and model adaptation is performed during the process operation without user interventions.

1.2 Bias-Variance dilemma

In the offline identification, the dilemma between bias and variance can be solved by determining a proper model complexity. The model complexity of a polynomial model structure can be adapted by the number of regressors through changing the polynomial degree or the allowed interaction. The complexity adaptation of neural network model structures can be performed by changing the number of regressors of the local models and additionally

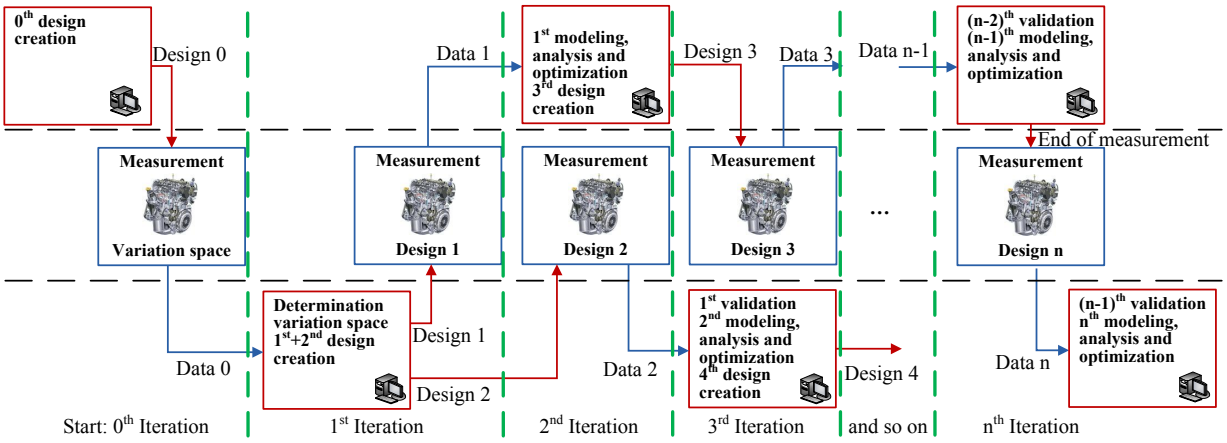

Fig. 1. Process of the iterative online measurements for combustion engine modeling with a parallelization of the measurement and modeling processes

through the model partition (e.g. number of local linear models of LOLIMOT, Nelles [2001]).

The bias-variance dilemma (see Fig.2a) states that a model with low complexity results in an underfitting solution (high bias error), yielding in high errors on both training and validation data. On the other hand, a model with a high complexity results in an overfitting solution (high variance error), yielding in low errors on training data but still high errors on validation data. Assuming a fixed data set size, a best complexity can be found regarding the resulting validation error by using information criteria like the Akaike information criterion or Mallows' C_p statistics (Miller [2002]).

In the introduced online measurements methodology, the sample size increases iteratively. The increasing data set size leads to different rules of thumb from a theoretical point of view to cope with the bias-variance dilemma. High variance can be fixed not only by simpler model structures, but additionally by more training data samples. Fixing the model structure, the bias error remains constant and the variance error decreases with the number of samples, resulting in an iteratively reduced total model error (see Fig.2b). Thus, in an online methodology, the total model error is automatically reduced by the increasing sample size. Further reduction of the total model error can be achieved by reduction of the bias error through adaptation of the model structure. The combination of both, the increasing sample size and adaptation of the model structure, yields to a reduction of the bias and variance error simultaneously, resulting in a higher model accuracy. Therefore, in an online methodology, not the best trade-off between bias and variance has to be found for a given data set, but furthermore the model structure has to be adapted target-oriented regarding the increasing number of data samples.

1.3 Local polynomial model trees

The *local polynomial model tree* (LOPOMOT, Sequenz [2013]) is the combination of an axis orthogonal splitting algorithm with adaptive polynomial models as local models. The local model regressors are selected iteratively with a stepwise selection algorithm, including or ignoring

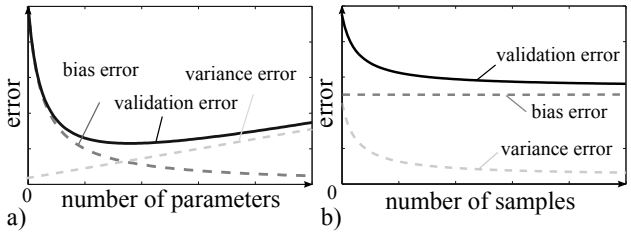

Fig. 2. Bias and variance for a) increasing number of parameters with a fixed data set size and b) increasing data set size with a fixed model complexity

regressors until the model error reaches a predefined limit. The qualities of the resulting different local models are compared, taking their complexities into account, with a local formulation of Mallows' C_p-statistic. The best model partition resulting from the tree construction algorithm is determined by a global formulation of Mallows' C_p-statistic, taking the model partitions M into account. The global model output $\hat{y}$ follows from the weighting of the resulting local models $\hat{y}_h$ with a pyramidal validity function $\Phi_h(\mathbf{z})$ to (see also Fig.3

$$\hat{y}(\mathbf{u}) = \sum_{h=1}^{M} \Phi_h(\mathbf{z})\hat{y}_h(\mathbf{u}). \tag{1}$$

2. ONLINE LOCAL POLYNOMIAL MODEL TREES

The offline LOPOMOT algorithm is designed for the identification of fixed data sets. During the online methodology, starting with only few data samples, new data samples are recorded iteratively and added to the identification data set. The real-time requirement (see section 1.1) enables the use of offline identification methods as long as the modeling does not take more time than the measurement. For modeling of nonlinear stationary and especially dynamic engine models, several hundred up to several thousand measurement samples have to be recorded and used for identification. The use of offline modeling methods can result in modeling times longer than measurement times, requiring the measurement to wait until the finish of the modeling. To avoid the resulting waiting times during the measurement and enable the online modeling,

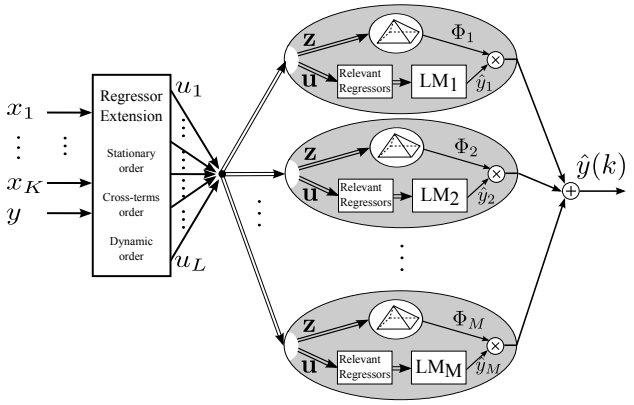

Fig. 3. LOPOMOT with adaptive local polynomial models that are weighted with a pyramidal weighting function to a global model output. The partition of the global model is determined by a tree construction algorithm.

the LOPOMOT procedure has to be adapted. *Online local polynomial model trees* (ONLOPOMOT) take into account the iteratively increasing data set and make use of recursive methods for parameter estimation, regressor selection and model partitioning instead of building a completely new model at every iterative modeling step. This speeds up the modeling procedure significantly and enables a good interaction with the measurement procedure.

2.1 Recursive online model analysis

In offline identification procedures, model analysis is done with mathematical measures like the coefficient of determination (R^2) or the root mean squared error (J_{RMSE}):

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}; \qquad J_{\mathrm{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{2}$$

Especially for model trees, quality criteria can be distinguished by global, local and global-local measures:

$$\begin{aligned} J_{\mathrm{RMSE,global}} &= \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \\ J_{\mathrm{RMSE,local}} &= \sqrt{\frac{\sum_{i=1}^{O}(y_i - \hat{y}_{h,i})^2}{O}} \\ J_{\mathrm{RMSE,global\text{-}local}} &= \sqrt{\frac{\sum_{i=1}^{O}(y_i - \hat{y}_i)^2}{O}} \end{aligned} \tag{3}$$

Hereby N stands for the number of total samples and O for the number of samples within the local model range. The local measure determines the quality of the local model ($\hat{y}_h$) within the local model range, while the local-global measure determines the quality of the global model ($\hat{y}$) within the local model range. Both take only the O data samples within the local model range into account. In contrast, the global measure determines the quality of the global model within the global model range with all available N data samples.

There are also normalized measures, but most of them use the mean output $\bar{y}$ to define the reference value. In the online identification, the mean values change with every new sample, changing the normalization factors. This makes the corresponding normalized analysis measures unsuitable for the decisions which local model to split and in which local model range to place new data samples to be measured. To enable the use of normalized criteria in the online methodology, factors independent of the mean output have to be used. This can be done either by normalization to the local (V_{LM}) and global (V_{GM}) model volume of the hyperquader

$$J_{\mathrm{NRMSEv}} = J_{\mathrm{RMSE}} \cdot \frac{V_{\mathrm{LM}}}{V_{\mathrm{GM}}} \tag{4}$$

or the normalization to the local and global model samples

$$J_{\mathrm{NRMSEs}} = J_{\mathrm{RMSE}} \cdot \frac{O}{N} \tag{5}$$

where V_{LM} and V_{GM} stand for the volume of the local and global hyperquader. Using this normalization, the quality remains the same for global models which consist of one local model with all measured samples. If there are splits present, the quality of small local models is rewarded in comparison to big ones. Using the normalization to samples, local models with more data samples are rewarded.

2.2 Recursive parameter adaptation

The parameters of the local nonlinear models, which are linear in their parameters, can be adapted with the recursive least squares (RLS) algorithm (Isermann and Münchhof [2011]) according to

$$\underbrace{\hat{\boldsymbol{\Theta}}(k+1)}_{\substack{\text{new}\\ \text{estimate}}} = \underbrace{\hat{\boldsymbol{\Theta}}(k)}_{\substack{\text{old}\\ \text{estimate}}} + \underbrace{\gamma(k)}_{\substack{\text{correction}\\ \text{vector}}} \underbrace{e(k+1)}_{\substack{\text{equation}\\ \text{error}}} \tag{6}$$

where the correction vector follows from

$$\gamma(k) = \frac{\mathbf{P}(k)\boldsymbol{\Psi}(k+1)}{\psi^T(k+1)\mathbf{P}(k)\psi(k+1)+1} \tag{7}$$

with the scaled estimate of the covariance matrix of the estimation error according to

$$\mathbf{P}(k+1) = \left(\mathbf{I} - \gamma(k)\psi^T(k+1)\right)\mathbf{P}(k) \tag{8}$$

and the initial values

$$\mathbf{P}(0) = \alpha\mathbf{I} \text{ and } \hat{\boldsymbol{\Theta}}(0) = 0. \tag{9}$$

The forgetting factor is omitted for the online methodology to gather as much information about the underlying process as possible. The advantage of using RLS within an online methodology is the fast and recursive calculation in comparison to ordinary least squares. The recursive parameter adaptation within ONLOPOMOT adapts only the parameters of the local models h, which are active for the new data samples $\mathbf{u}(k+1)$

$$\Phi_h(\mathbf{u}(k+1)) \geq 0.5. \tag{10}$$

By this way, an unlearning of the properties of the inactive models is avoided (Nelles [2001]).

2.3 Recursive model partition adaptation

Initial structure split The offline modeling LOPOMOT algorithm for model partition separates the regressor space into hyperquaders by iteratively splitting the worst local model with a fixed splitting ratio. As the initial net is generated based on the present data, the regressor space is fully spanned between $\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}$ in the offline case as all data is present. In the online case, only few data $\mathbf{u}_{\mathrm{i}}$

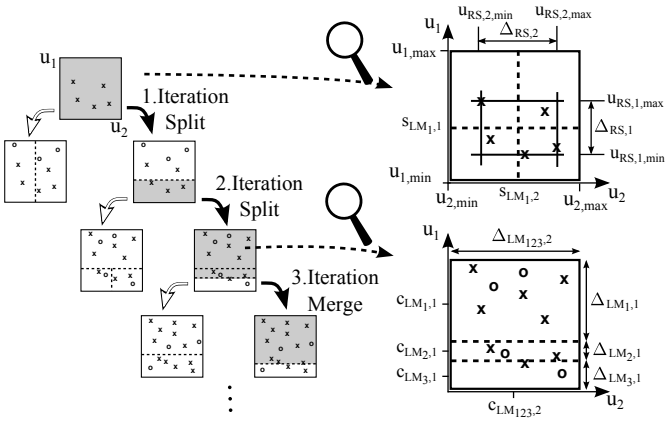

Fig. 4. Axis orthogonal splitting and merging algorithm of ONLOPOMOT

with $i = 1 \dots S$ and $S \ll N$ is available at the beginning with

$$\mathbf{u}_{\min} \ll \mathbf{u}_{\mathrm{i}} \ll \mathbf{u}_{\max}. \tag{11}$$

Hereby S is the number of initial data samples. The bigger the distances of $\mathbf{u}_{\mathrm{i}}$ to $\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}$ are, the smaller is the initially spanned regressor variation space, leading to a global model with a small interpolation and big extrapolation range after the end of the modeling. To avoid the extrapolation behavior, the initial regressor space of ONLOPOMOT is not spanned by the data, but by the given limits of the measured variables $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ (see Fig.4).

Splitting ratio adaptation Another problem arises with the data capturing procedure. The measured data can't be guaranteed to be equally distributed. As shown in Fig.4, iteratively captured data can be focused in a limited range of the global model, making a split with a fixed ratio, e.g. $r_{\mathrm{split}} = 0.5$, impossible due to the lack of data. Therefore the ONLOPOMOT algorithm adapts the given fixed split ratio for each dimension g to the range of all relevant samples ($\Delta_{RS,g}$) within the local model LM (see Fig.4) by

$$r^*_{\mathrm{split,g}} = \Delta_{RS,g} \cdot r_{\mathrm{split,g}} + u_{\mathrm{RS,g,min}} \tag{12}$$

resulting in an individual adaptation of the splitting ratio in each iteration to the present data samples. The splits $s_{LM_h,g}$ of the local model h follow to

$$s_{LM_h,g} = \Delta_{LM_h,g} \cdot r^*_{\mathrm{split,g}} \tag{13}$$

Splitting dimension determination The LOPOMOT algorithm splits the local model in all possible dimensions, compares the resulting global quality criteria J_{global} and inherits the split with the highest resulting quality. Again, under the constraint of iteratively added data samples which are not equally distributed, the criterion has to be adapted. To avoid an iteratively repeated splitting in the same dimension due to a lack of data distribution, the global criterion is additionally weighted with the hyperquader side lengths of each dimension:

$$J_{\mathrm{RMSEd,global}} = J_{\mathrm{RMSE,global}} \cdot \frac{\sum_{g=1}^{L} \Delta_{LM_h,g}}{\Delta_{LM_h,d}}. \tag{14}$$

This results in the decrease of global model qualities where splits are made in dimensions with small side lengths, resulting in favored splits in dimensions with longer side lengths.

Structure merging In the case of iteratively measured data, not only the splitting of the model tree but also the merging becomes relevant. With new data, a former split can become obsolete. Therefore, the ONLOPOMOT algorithm not only includes a splitting, but also a merging algorithm. Junge [1999] defined requirements for merging of local linear model trees, which are adapted for online local polynomial model trees:

(1) *The merged local model LM_m has to be a hyperquader again.* This can be accomplished by comparing the side lengths of two local hyperquaders $\Delta_{LM_k,g}$ and $\Delta_{LM_l,g}$ in each dimension $g = 1 \dots L$. Only the to be merged side lengths of the dimension d are allowed to differ from each other, all other side lengths have to be equal:

$$\Delta_{LM_k,g} \stackrel{!}{=} \Delta_{LM_l,g} \text{ with } \quad g \neq d \tag{15}$$

(2) *The to be merged local models LM_k and LM_l have to be next to each other.* This can be achieved by ensuring the following equation for the centers of the to be merged local models:

$$|c_{LM_k,d} - c_{LM_l,d}| \stackrel{!}{=} \frac{1}{2}[\Delta_{LM_k,d} + \Delta_{LM_l,d}] \tag{16}$$

Additionally all other center dimensions $g = 1 \cdots L$ have to be equal:

$$c_{LM_k,g} \stackrel{!}{=} c_{LM_l,g} \text{ with } \quad g \neq d \tag{17}$$

Otherwise, the center c_{LM_m} of the new local model would be in a non active region and the local model LM_m would never have a high activity.

(3) *The local model parameters have to be similar.* This can be guaranteed by limits on parallelism (Δ_{angle}) and origin deviation (Δ_{offset}). Parallelism can be determined by the angle between the normal vectors of the local model parameter hyperplanes

$$\arccos\left(\frac{\hat{\mathbf{\Theta}}_k^T \hat{\mathbf{\Theta}}_l}{|\hat{\mathbf{\Theta}}_k||\hat{\mathbf{\Theta}}_l|}\right) \stackrel{!}{\leq} \Delta_{\mathrm{angle}} \tag{18}$$

and origin deviation by the difference between the offsets

$$|w_{k,0} - w_{l,0}| \stackrel{!}{\leq} \Delta_{\mathrm{offset}}. \tag{19}$$

Hereby $\hat{\mathbf{\Theta}}_k$ is the parameter vector $[w_{k,0}, ..., w_{k,L}]^T$ and L the number of total regressors without the offset.

Applying this three requirements, local models can be found, which are supposed to be merged. The new resulting local model can be estimated by determining the relevant regressors and calculation of the new parameters.

2.4 Recursive regressor selection

The stepwise selection algorithm of the offline LOPOMOT algorithm has the disadvantage in the online modeling, that new incoming samples include new process information, making an old selection procedure inappropriate. In each iteration, the stepwise selection would have to be started over again, leading to long training times in each iteration. Furthermore, the stepwise selection based on Mallows' C_p-statistic is strongly correlated to the number of samples, increasing when applied to increasing sample sizes. In an online modeling algorithm, the regressor selection should start already when only few samples are

present and adapt to the incoming samples. Therefore, the ONLOPOMOT algorithm uses a combination of the Lasso algorithm (Hastie et. al [2009]) and a local formulation of the corrected AIC criterion (Miller [2002]) to select the significant regressors of the local polynomial models.

Lasso is a shrinkage and selection method for regression by minimizing the sum of squared errors while putting a bound on the sum of the absolute values of the parameters $\hat{\mathbf{\Theta}}$ (Hastie et. al [2009]). The resulting objective function, which has to be minimized, follows to

$$J_\lambda(\hat{\mathbf{\Theta}}) = \sum_{i=1}^{N} \left(\mathbf{y}_i - \mathbf{X}_i\hat{\mathbf{\Theta}}\right)^2 + \lambda \sum_{g=0}^{L} \left|\hat{\mathbf{\Theta}}_g\right| \qquad (20)$$

for different $\lambda > 0$. The difficulty to find an appropriate λ to weight the bound is overcome in ONLOPOMOT by comparing the resulting local models for different λ values by the corrected AIC criterion:

$$\mathrm{AIC_c} = N \cdot \log\left(\frac{\mathrm{RSS}}{N}\right) + 2L_{\mathrm{eff}} + \frac{2L_{\mathrm{eff}}\left(L_{\mathrm{eff}}+1\right)}{N - L_{\mathrm{eff}} - 1} \qquad (21)$$

Here RSS stands for the estimated residual sum of squares, N the number of total samples and L_{eff} the effective number of parameters of the fitted local model. The corrected AIC criterion, in comparison to the normal AIC criterion or Mallows' $\mathrm{C_p}$-statistic, is suitable for small sample sizes relative to the number of parameters ($N/L_{\mathrm{eff}} < 40$) and generally suitable unless the probability distribution is extremely non-normal. For $N \to \infty$ the corrected AIC approaches the normal AIC criterion (Miller [2002]).

The use of the Lasso combined with a λ selection by the $\mathrm{AIC_c}$ enables an adaptation of the local model regressors to the non-linearities of the process during the online modeling even with small sample sizes. It should be mentioned here, that the $\mathrm{AIC_c}$ criterion is only used for finding the appropriate λ, and not for comparing different local models among each other because the $\mathrm{AIC_c}$, as well as other information criteria like AIC or Mallows' $\mathrm{C_p}$-statistic are not suitable to compare models with different sample sizes.

2.5 Online identification methodology

The resulting methodology of the ONLOPOMOT algorithm is presented in Fig.5.

With the first sampled data, an initial model is identified, consisting of one local model. Every time, a measuring iteration is finished and the newly sampled data is added to the data set, the following steps are performed:

(1) The global validation error (e.g. root mean square error) for the S new samples is determined:

$$J_{\mathrm{RMSE,global}} = \sqrt{\frac{\sum_{i=1}^{S}(y_i - \hat{y}_{\mathrm{global,i}})^2}{S}} \qquad (22)$$

As long as the validation error is below a specified threshold G_{global}, no model adaptation is performed and new data is iteratively sampled.

(2) When the validation error $J_{\mathrm{RMSE,global}}$ for a new sample rises above the threshold, an adaptation of the active local model parameters is performed with the RLS for the new data samples. After the adaptation of the local model parameters, the global-local model error $J_{\mathrm{RMSE,global\text{-}local}}$ is evaluated for all active local models. If the error is below the specified threshold $G_{\mathrm{RMSE,global\text{-}local}}$, the data sampling continues.

(3) If the error of the active local model is above the specified threshold $G_{\mathrm{global\text{-}local}}$, a regressor selection is performed, with the Lasso algorithm starting at $\lambda(iter-1)$. After the regressor selection, the local-global error of all present (and not only the active ones) local models is evaluated. If none of the local model has an error above the threshold, sampling continues.

(4) Otherwise, the local model with the highest volume normalized error $J_{\mathrm{NRMSEv,global\text{-}local}}$ above the threshold is split according to section 2.3. Using the normalization to the volume, the quality of small local models is rewarded, resulting in a preferred split in bigger local models.

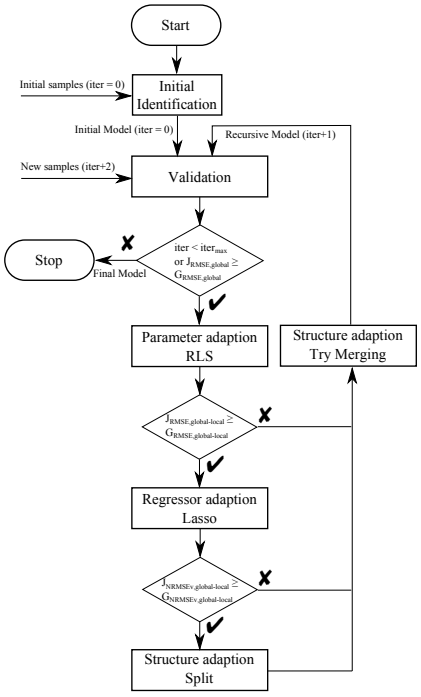

Fig. 5. Methodology of the ONLOPOMOT algorithm

Additionally, after each adaptation of the model and before new sampling, a merging of the local models is tried. If possible, local models are merged, resulting in a global model with a reduced global complexity.

To combine the measuring and modeling process, new data is sampled within the worst local models. The quality is determined by the sample normalized error $J_{\mathrm{NRMSEs,global\text{-}local}}$ of the global model in the local model ranges. The use of the sample normalized error results in a preferred placing of new to be measured data samples in local model ranges with less present data samples.

3. EXAMPLES FOR ONLINE IDENTIFICATION

3.1 Online modeling of an exemplary stationary function

An exemplary function of two variables u_1 and u_2 is shown in Fig.6a), consisting of several scaled gaussian distribu-

tions. The modeling is performed for 200 iterations and the model error determined as J_{RMSE}. The initial model is generated out of 9 data samples, each consisting of a combination of values of the input and output variables (x_1, x_2, y). In each iteration, 2 new random data samples are generated, one in each of the two worst local models, resulting in a total of 409 recorded samples (see Fig.6b)). During regressor selection the maximum possible stationary order of the input variables is three (x_K, x_K^2, x_K^3) and up to two cross-terms of the input variables are allowed as regressors $(x_1 \cdot x_2)$.

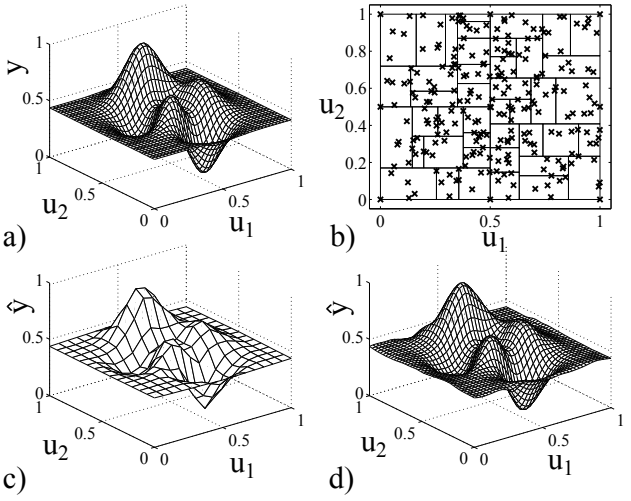

Fig. 6. Stationary online modeling with ONLOPOMOT of a test function. a) Test function to be modeled. b) Local model parition of ONLOPOMOT and sampled data points after end of modeling. c) Training result for modeled test function. d) Validation result for modeled test function

During the online modeling, 160 samples are ignored due to good validity, 120 times the parameters of the local models are adapted, 114 times the regressors updated and 42 times the model is split. The final model consists of 43 local models, where each has different relevant regressors and parameters. The resulting stationary ONLOPOMOT after the end of the measurement is shown in Fig.6c). The gaussian test function can be modeled rather good, with an error of $J_{\mathrm{RMSE,global}} = 0.0061$. Additionally, in Fig.6d), the validation with a grid (40x40) is shown. The validation error is also low with $J_{\mathrm{RMSE,global}} = 0.0087$. In comparison to ONLOPOMOT, using the offline LOPOMOT algorithm and the same data samples, the training error is higher with $J_{\mathrm{RMSE,global}} = 0.071$. As shown in Fig.6b), the model split and merge algorithm of ONLOPOMOT focuses splits in regions of stronger non-linearity, although it performs iteratively. Additionally, the online methodology combines the splitting of the model and placing of new to be measured data samples, increasing the data samples density in region of high non-linearity, enabling a significant increase in modeling quality.

As shown in Fig.7a), the training error decreases in general during the iterative procedure. Additionally, also the validation error of the iteratively added samples (b) as well as the validation error of a separate data set (c) decreases in general with the ongoing online measurement

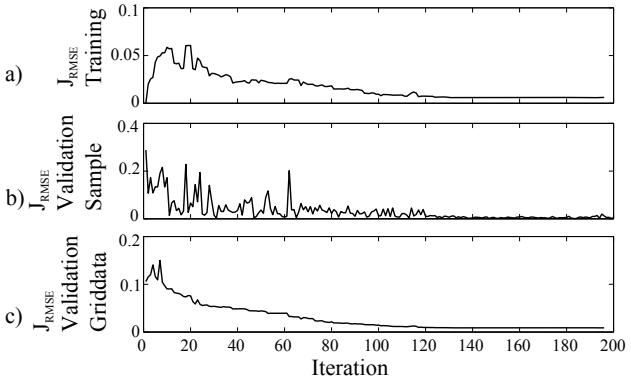

Fig. 7. Quality criterions during online modeling iterations. The training error (a), the validation error of the new sampled data points (b) and the validation error of a validation data set (c) decrease continuously during the iterations.

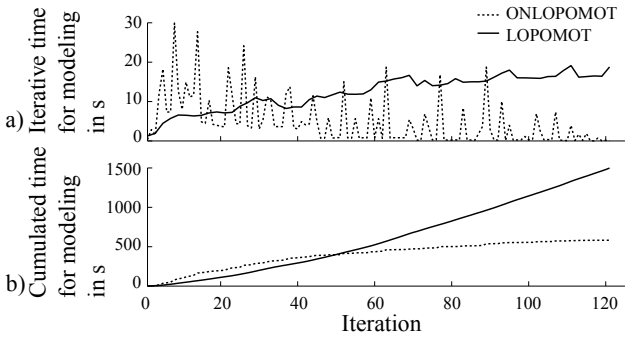

Fig. 8. Iterative (a) and cumulated (b) time needed for the LOPOMOT and ONLOPOMOT modeling iterations. The break-even point for the online modeling occurs at iteration.

and modeling. Therefore, as already mentioned in section 1.2, the online modeling with the presented ONLOPOMOT algorithm enables the reduction of the model bias and variance, resulting in models of high qualities.

Another advantage of the ONLOPOMOT algorithm is the time needed for modeling. Fig.8 shows the iterative and cumulated time for the online and offline algorithms. The LOPOMOT algorithm needs more time with every new iteration, as the data set size steadily increases. On the other hand, the iterative time needed in the ONLOPOMOT algorithm even decreases, as adaptations (parameter, regressor and structure adaptation) are only made for relevant samples and data samples not contributing much to the modeling are ignored. The cumulated time shows the resulting break-even point at iteration 49 (107 samples), where the total time needed of the offline modeling exceeds the online modeling time. This example shows the capability of the presented methodology, to reduce the time needed for modeling and therefore reduce the overall time needed for the measurements procedure.

3.2 Online modeling of the diesel engine boost pressure

As a second example, the ONLOPOMOT methodology is used for modeling of the dynamic boost pressure p_2. Input variables are the positions of the exhaust gas recirculation (egr) valve and the variable geometry turbine (vgt) of

the turbocharger at the operating point 2000rpm and 10mm^3/cyc. Both are excited dynamically with steps of 5s length and amplitudes chosen in local model regions of worst model quality (Kowalczyk [2013]). The step length is chosen to be longer than the settling time to enable the "'excitation"' of the stationary and dynamic process behavior. During regressor selection the maximum possible stationary order of the input variables is five $(x_K, ..., x_K^5)$ and delays with up to five samples can be selected as dynamic regressors $(x_K(k), x_K(k-1), ..., x_K(k-5))$, resulting in a nonlinear dynamic model. In Fig.9 the input variables, resulting output variable and model output as well as the design are shown.

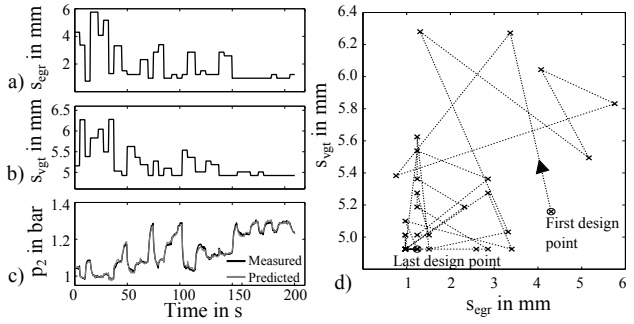

Fig. 9. Dynamic online modeling with ONLOPOMOT of the boost pressure. a)+b) Time plot of the modeling inputs s_{egr} and s_{vgt}. c) Measured and predicted output p_2. d) Corresponding measurement design.

In contrast to the stationary case, measured data points can't be ignored during modeling. As shown in Fig.9d), removing of single design points from the data would intersect the progress of the variables, resulting in non-consistet modeling data. Therefore all data is used for modeling. The resulting dynamic ONLOPOMOT after the end of the measurement is shown in Fig.9c). In total 420 samples (sampling time $T_0 = 0.5s$) in 7 iterations were measured before the final model quality was reached. The boost pressure is modeled with a quality of $J_{\text{RMSE,global}} = 0.0082$. The validation error, determined with a seperate space filling design, is also good with $J_{\text{RMSE,global}} = 0.0234$. In Fig.9d) again the focusing of the ONLOPOMOT algorithm in relevant regions can be seen. This example shows the capability of the presented methodology, to determine iteratively the proper model structure after only few measurements and therefore reduces the needed amount of measurements.

4. CONCLUSION AND OUTLOOK

A new methodology for online determination of stationary and dynamic combustion engine models at test benches was developed. In comparison to common approaches, the steps of measurement are performed autonomously one after the other, without any separation of measurement at test benches and offline processing at the office. Furthermore, this approach considers the measured data for adaptation of the model structure.

First the online iterative methodology was introduced and described. The developed methodology uses real-time hardware (online real-time measurement) parallel to an office computer (online processing), enabling the use of offline identification methods in the online methodology. The necessary extensions of the modeling process were shown. To cope with the increasing sample size during the iterative online modeling, recursive methods have to be used for model analysis, parameter adaptation, regressor selection and model partition adaptation. All this extensions were merged in the online local polynomial model tree methodology (ONLOPOMOT), which is capable of recursive online modeling of nonlinear processes. The applicability of the resulting methodology was shown for two emaples, the modeling of an simulated exemplary stationary test function and the modeling of the dynamic boost pressure of an engine at a test bench with two input variables.

The implemented method enables a reduction of the measurement and modeling expense by reducing the time needed for modeling and measurement. Future research is oriented in the target oriented placing of iterative measurement points. So far, random samples were generated in worst local model regions. By using active learning methods (Kowalczyk [2014]), new to be measured data samples can be placed where the highest relevance for the modeling is.

ACKNOWLEDGEMENTS

This contribution is the scientific result of a research project, which was supported by the Forschungsvereinigung Verbrennungskraftmaschinen e.V. (FVV, Frankfurt) and worked out by the Institute of Automatic Control and Mechatronics at the TU Darmstadt. The project was financed by the Federal Ministry of Economy and Technology (BMWi) over the German Federation of Industrial Research Associations (AiF, Nr.16400 and Nr.17611).

REFERENCES

T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics, Second Edition, New York, 2009.

R. Isermann, M. Münchhof *Identification of Dynamic Systems: An Introduction with Applications* . Springer, 1.edition , 2011.

T.F. Junge. *'On-line'-Identifikation und lernende Regelung nichtlinearer Regelstrecken mittels neuronaler Netze.* Dissertation, Bochum, 1999.

M. Kowalczyk. *Online methods I: Optimized fast engine measurements with online-methods for determination of stationary and dynamic models.* FVV project report, Frankfurt, 2013.

M. Kowalczyk. *Online methods II: Extension of online measurements by active learning optimization for emission and consumption reduction.* FVV project report, Frankfurt, 2014.

A. Miller. *Subset Selection in Regression.* Monographs on Statistics and Applied Probability. CRC Press, Floriada, 2002.

O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models.* Springer, 1.edition , 2001.

H. Sequenz. *Emission Modelling and Model-Based Optimisation of the Engine Control.* Dissertation, TU Darmstadt, 2013.